

LU decomposition

From Wikipedia, the free encyclopedia

In linear algebra, the **LU decomposition** is a matrix decomposition which writes a matrix as the product of a lower and upper triangular matrix. The product sometimes includes a permutation matrix as well. This decomposition is used in numerical analysis to solve systems of linear equations or find the inverse of a matrix.

Contents

- 1 Definitions
- 2 Existence and uniqueness
- 3 Positive definite matrices
- 4 Algorithms
 - 4.1 Doolittle algorithm
 - 4.2 LUP algorithm
- 5 Small Example
- 6 Applications
 - 6.1 Solving linear equations
 - 6.2 Inverse matrix
- 7 See also
- 8 References
- 9 External links

Definitions

Let A be a square matrix. An *LU decomposition* is a decomposition of the form

$$A = LU,$$

where L and U are lower and upper triangular matrices (of the same size), respectively. This means that L has only zeros above the diagonal and U has only zeros below the diagonal. For a 3×3 matrix, this becomes:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

An *LDU decomposition* is a decomposition of the form

$$A = LDU,$$

where D is a diagonal matrix and L and U are *unit* triangular matrices, meaning that all the entries on the diagonals of L and U are one.

EXHIBIT A

A *LUP decomposition* is a decomposition of the form

$$A = LUP,$$

where L and U are again lower and upper triangular matrices and P is a permutation matrix, i.e., a matrix of zeros and ones that has exactly one entry 1 in each row and column.

Existence and uniqueness

An invertible matrix admits an LU factorization if and only if all its principal minors are non-zero. The factorization is unique if we require that the diagonal of L (or U) consist of ones. The matrix has a unique LDU factorization under the same conditions.

If the matrix is singular, then an LU factorization may still exist. In fact, a square matrix of rank k has an LU factorization if the first k principal minors are non-zero.

The exact necessary and sufficient conditions under which a not necessarily invertible matrix over any field has an LU factorization are known. The conditions are expressed in terms of the ranks of certain submatrices. The Gaussian elimination algorithm for obtaining LU decomposition has also been extended to this most general case (Okunev & Johnson 1997).

Every matrix A --square or not-- admits a LUP factorization. The matrices L and P are square matrices, but U has the same shape as A . *Upper triangular* should be interpreted as having only zero entries below the main diagonal, which starts at the upper left corner. The LUP factorization can be done in such a way that U has only ones on its main diagonal.

Positive definite matrices

If the matrix A is Hermitian and positive definite, then we can arrange matters so that U is the conjugate transpose of L . In this case, we have written A as

$$A = LL^*.$$

This decomposition is called the Cholesky decomposition. The Cholesky decomposition always exists and is unique. Furthermore, computing the Cholesky decomposition is more efficient and numerically more stable than computing the LU decomposition.

Algorithms

The LU decomposition is basically a modified form of Gaussian elimination. We transform the matrix A into an upper triangular matrix U by eliminating the entries below the main diagonal. The Doolittle algorithm does the elimination column by column starting from the left, by multiplying A to the left with atomic lower triangular matrices. It results in a *unit lower triangular* matrix and an upper triangular matrix. The Crout algorithm is slightly different and constructs a lower triangular matrix and a *unit upper triangular* matrix.

EXHIBIT A

Computing the LU decomposition using either of these algorithms requires $2n^3/3$ floating point operations, ignoring lower order terms. Partial pivoting adds only a quadratic term and can thus be neglected; this is not the case for full pivoting (Golub & Van Loan 1996).

Doolittle algorithm

Given an $N \times N$ matrix

$$A = (a_{n,n})$$

we define

$$A^{(0)} := A$$

and then we iterate $n = 1, \dots, N-1$ as follows.

We eliminate the matrix elements below the main diagonal in the n -th column of $A^{(n-1)}$ by adding to i -th row of this matrix the n -th row multiplied by

$$l_{i,n} := -\frac{a_{i,n}^{(n-1)}}{a_{n,n}^{(n-1)}}$$

for $i = n+1, \dots, N$. This can be done by multiplying $A^{(n-1)}$ to the left with the lower triangular matrix

$$L_n = \begin{pmatrix} 1 & & & & 0 \\ & \ddots & & & \\ & & 1 & & \\ & & l_{n+1,n} & \ddots & \\ & & \vdots & \ddots & \\ 0 & & l_{N,n} & & 1 \end{pmatrix}.$$

We set

$$A^{(n)} := L_n A^{(n-1)}.$$

After $N-1$ steps, we eliminated all the matrix elements below the main diagonal, so we obtain an upper triangular matrix $A^{(N-1)}$. We find the decomposition

$$A = L_1^{-1} L_1 A^{(0)} = L_1^{-1} A^{(1)} = L_1^{-1} L_2^{-1} L_2 A^{(1)} = L_1^{-1} L_2^{-1} A^{(2)} = \dots = L_1^{-1} \dots$$

EXHIBIT A

Denote the upper triangular matrix $A^{(N-1)}$ by U , and $L = L_1^{-1} \dots L_{N-1}^{-1}$. Because the inverse of a lower triangular matrix L_n is again a lower triangular matrix, and the multiplication of two lower triangular matrices is again a lower triangular matrix, it follows that L is a lower triangular matrix. We obtain $A = LU$.

It is clear that in order for this algorithm to work, one needs to have $a_{n,n}^{(n-1)} \neq 0$ at each step (see the definition of $L_{i,n}$). If this assumption fails at some point, one needs to interchange n -th row with another row below it before continuing. This is why the LU decomposition in general looks like $P^{-1}A = LU$.

LUP algorithm

The LUP decomposition algorithm by Cormen et al. generalizes Crout matrix decomposition. It can be described as follows.

1. If A has a nonzero entry in its first row, then take a permutation matrix P_1 such that AP_1 has a nonzero entry in its upper left corner. Otherwise, take for P_1 the identity matrix. Let $A_1 = AP_1$.
2. Let A_2 be the matrix that one gets from A_1 by deleting both the first row and the first column. Decompose $A_2 = L_2U_2P_2$ recursively. Make L from L_2 by first adding a zero row above and then adding the first column of A_1 at the left.
3. Make U_3 from U_2 by first adding a zero row above and a zero column at the left and then replacing the upper left entry (which is 0 at this point) by 1. Make P_3 from P_2 in a similar manner and define $A_3 = A_1 / P_3 = AP_1 / P_3$. Let P be the inverse of P_1 / P_3 .
4. At this point, A_3 is the same as LU_3 , except (possibly) at the first row. If the first row of A is zero, then $A_3 = LU_3$, since both have first row zero, and $A = LU_3P$ follows, as desired. Otherwise, A_3 and LU_3 have the same nonzero entry in the upper left corner, and $A_3 = LU_3U_1$ for some upper triangular square matrix U_1 with ones on the diagonal (U_1 clears entries of LU_3 and adds entries of A_3 by way of the upper left corner). Now $A = LU_3U_1P$ is a decomposition of the desired form.

Small Example

$$\begin{bmatrix} 4 & 3 \\ 6 & 3 \end{bmatrix} = \begin{bmatrix} l_{11} & 0 \\ l_{21} & l_{22} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{bmatrix}$$

One way of finding the LU decomposition of this simple matrix would be to simply solve the linear equations by inspection. You know that:

$$l_{11} * u_{11} + 0 * 0 = 4$$

$$l_{11} * u_{12} + 0 * u_{22} = 3$$

$$l_{21} * u_{11} + l_{22} * 0 = 6$$

$$l_{21} * u_{12} + l_{22} * u_{22} = 3$$

EXHIBIT A

Such a system of equations is underdetermined. In this case any two non-zero elements of L and U matrices are parameters of the solution and can be set arbitrarily to any non-zero value. Therefore to find the unique LU decomposition, it is needed to put some restriction on L and U matrices. For example, we can require the upper triangular matrix U to be a unit one (i.e. set all the entries of its main diagonal to ones). Then the system of equations has the following solution:

$$\begin{aligned}l_{11} &= 4 \\u_{12} &= 0.75 \\l_{21} &= 6 \\l_{22} &= -1.5\end{aligned}$$

Substituting these values into the LU decomposition above:

$$\begin{bmatrix} 4 & 3 \\ 6 & 3 \end{bmatrix} = \begin{bmatrix} 4 & 0 \\ 6 & -1.5 \end{bmatrix} \begin{bmatrix} 1 & 0.75 \\ 0 & 1 \end{bmatrix}$$

Applications

Solving linear equations

Given a matrix equation

$$Ax = LUx = b$$

we want to solve the equation for a given A and b . In this case the solution is done in two logical steps:

1. First, we solve the equation $Ly = b$ for y
2. Second, we solve the equation $Ux = y$ for x .

Note that in both cases we have triangular matrices (lower and upper) which can be solved directly using forward and backward substitution without using the Gaussian elimination process (however we need this process or equivalent to compute the LU decomposition itself). Thus the LU decomposition is computationally efficient only when we have to solve a matrix equation multiple times for different b . It is faster to do a LU decomposition of the matrix A once and then solve the triangular matrices for the different b than to use Gaussian elimination each time.

Inverse matrix

The matrices L and U can be used to calculate the matrix inverse.

Computer implementations that invert matrices often use this approach.

See also

- Block LU decomposition

EXHIBIT A

- Cholesky decomposition

References

- Cormen, T.H.; C.E Leisserson & R.L. Rivest, *Introduction to Algorithms*
- Golub, Gene H. & Charles F. Van Loan (1996), *Matrix Computations* (3rd ed.), Baltimore: Johns Hopkins, ISBN 978-0-8018-5414-9.
- Horn, Roger A. & Charles R. Johnson (1985), *Matrix Analysis*, Cambridge University Press, ISBN 0-521-38632-2. See Section 3.5.
- Okunev, Pavel & Charles R. Johnson (1997), *Necessary And Sufficient Conditions For Existence of the LU Factorization of an Arbitrary Matrix*, arXiv:math.NA/0506382 (<http://arxiv.org/abs/math.NA/0506382>).
- LU decomposition (<http://mathworld.wolfram.com/LUDecomposition.html>) on *MathWorld*.
- LU decomposition (<http://www.math-linux.com/spip.php?article51>) on *Math-Linux*.

External links

- LAPACK (<http://www.netlib.org/lapack/>) is a collection of FORTRAN subroutines for solving dense linear algebra problems
- ALGLIB (<http://www.alglib.net/>) includes a partial port of the LAPACK to C++, C#, Delphi, etc.
- Online Matrix Calculator (<http://www.bluebit.gr/matrix-calculator/>) performs LU decomposition
- LU decomposition (http://numericalmethods.eng.usf.edu/mws/gen/04sle/mws_gen_sle_txt_ludecomp.doc) at *Holistic Numerical Methods Institute*
- Module for LU Factorization with Pivoting (<http://math.fullerton.edu/mathews/n2003/LUFactorMod.html>)
- LU Decomposition (<http://demonstrations.wolfram.com/LUDecomposition/>) at the Wolfram Demonstrations Project

Retrieved from "http://en.wikipedia.org/wiki/LU_decomposition"

Category: Numerical linear algebra

- This page was last modified 22:26, 23 June 2007.
- All text is available under the terms of the GNU Free Documentation License. (See **Copyrights** for details.) Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a US-registered 501(c)(3) tax-deductible nonprofit charity.

EXHIBIT A